Achieving Correctness in Fair Rational Secret Sharing

Sourya Joyee De and Asim K. Pal

Management Information Systems Group,
Indian Institute of Management Calcutta, India

Abstract. In rational secret sharing, parties may prefer to mislead others in believing a wrong secret as the correct one over everybody obtaining the secret (i.e. a fair outcome). Prior rational secret reconstruction protocols for non-simultaneous channel only address the case where a fair outcome is preferred over misleading and hence are fair but not correct. Asharov and Lindell (2010) proposed the first and the only protocol that takes care of both the preferences. In this paper, we propose a new rational secret sharing protocol that addresses both the preferences and is fair and correct in the non-simultaneous channel model. Additionally, it is independent of the utility of misleading. Each rational party is given a list of sub-shares of shares of the actual secret and fake shares. In each round of the protocol each party sends the current element in its list to the other party and then reconstructs a share from the sub-shares obtained. The main idea is to use a checking share which is a share of the original secret as a protocol–induced membership auxiliary information to check whether the shares obtained till a certain round can be used to reconstruct the correct secret. We overcome the disadvantages of the presence of auxiliary information by using the time-delayed encryption scheme used by the protocol of Lysyanskaya and Segal (2010) that tolerates players with arbitrary side information. In our case, the side information used is not arbitrary but introduced by the mechanism/protocol designer to put all players on equal footing. We show that our protocol is in computational strict Nash equilibrium in the presence of protocol-induced auxiliary information.

1 Introduction

Since the introduction of the concept of rational players in (t, n) threshold secret sharing by [6], the area which henceforth came to be known as rational secret sharing (RSS) and its application in secure multiparty computation (known as rational multi-party computation or RMPC) has attracted a lot of fruitful research [1, 2, 5, 7–9, 13, 15–17, 19–21]. Briefly, the RSS problem is as follows. Each of n players $P_1, P_2, \ldots, P_n$ is given a share of a secret s by a dealer. The secret can be reconstructed if at least any t of them cooperate. However, the point of contention is that each player wishes to learn the secret himself while allowing as few others as possible to learn the correct value. What strategy will each player need to adopt so that each player comes to know the secret?

M. Abdalla, C. Nita-Rotaru, and R. Dahab (Eds.): CANS 2013, LNCS 8257, pp. 139–161, 2013.

Inherent to the RSS problem is the problem of achieving fairness. Each player wants to obtain the secret alone and is unfair to others i.e. for each player, the utility of obtaining the secret alone (U^{TN}) is the maximum. The other utilities of a rational party are that of everybody obtaining the secret (U^{TT}), that of nobody obtaining the secret (U^{NN}) and that of everybody else obtaining the secret (U^{NT}). So each player has a preference of $U^{TN} > U^{TT} > U^{NN} > U^{NT}$. The desirable outcome of the secret reconstruction game is the fair one in which everybody obtains the secret. A rational secret reconstruction scheme or protocol is a strategy for each player suggested by the protocol designer such that this fair outcome can be obtained and there is no incentive for any player to deviate from this strategy. Nash equilibrium and its variants (computational Nash, strict Nash etc) are the most used equilibrium concepts in this context. Much of the RSS literature [2, 6, 15, 21] focusses on obtaining fair rational secret reconstruction mechanisms under different assumptions such as the type of communication channel present (simultaneous/ non-simultaneous) or the nature of the dealer (online/offline). We present a brief comparative summary of such protocols in Table 1. The basic assumption about the preference $U^{TN} > U^{TT} > U^{NN} > U^{NT}$ of rational players is common to all the RSS protocols proposed so far (hence, we do not mention this separately in Table 1). In some cases, there are some special assumptions (which we mention in Table 1, under 'Special Preferences ') about the nature of players (for eg., [16] assumes a rational majority along with a minority of honest players) and their preferences. These special preferences are related to the correctness of the secret obtained ([2]).

Parties in a rational secret reconstruction mechanism may often be considered to derive some positive utility from misleading other players into believing a wrong value to be the correct secret when it itself obtains nothing (U^{NF}). A fair reconstruction protocol gives the utility of U^{TT} to each player. Therefore it is also correct as long as $U^{NF} < U^{TT}$. However, when parties prefer misleading others over everybody obtaining the correct secret (i.e. $U^{NF} \geq U^{TT}$), a fair rational secret reconstruction protocol for the non-simultaneous channel model does not remain correct (we shall soon discuss why this is so). Unfortunately, this problem has received very little attention from researchers and this can be easily identified from Table 1. [2] proposed the first and the only correct and fair rational secret reconstruction protocol for the case when both scenarios may hold in the $(2, 2)$ setting. Prior to their work, all works on rational secret sharing either assumed the existence of simultaneous broadcast channel [5, 6] (where this problem does not exist) or assumed that rational parties prefer everybody to obtain the output of the protocol than misleading others [9, 15, 21]. We therefore aim to design a correct and fair rational secret reconstruction protcol in the non-simultaneous channel.

A desirable property of any rational secret reconstruction scheme is utility-independence. If a particular RSS scheme is dependent on utility values of players then it requires the protocol designer to be able to accurately estimate the utility values or at least the range of these values in order to set the appropriate parameters during the execution of that RSS scheme. The work of [2] has extensively dealt with the property of utility-independence. It proposes a (t, n) (where $n \geq 3$,

$2 < t \leq n$) rational secret reconstruction protocol which is completely utility-independent (i.e. the protocol designer is not required to know any utility value) in the simultaneous channel model. However, [2] also showed that, in the non-simultaneous channel model, there does not exist any $(2, 2)$ fair rational secret reconstruction protocol that is independent of the utility value U^{NF}. Consequently, the $(2, 2)$ correct and fair rational secret reconstruction protocol they suggest in the non-simultaneous channel, although correct even when $U^{NF} \geq U^{TT}$, is U^{NF} utility–dependent. In this paper we remove this utility dependency. So the basic question that we address here is whether it is possible to have a rational secret reconstruction protocol that is both correct and fair even when $U^{NF} \geq U^{TT}$ and given that, whether it is possible to achieve U^{NF}–independence for such a protocol. We propose a $(2, 2)$ fair rational secret reconstruction mechanism in the non-simultaneous channel that is 1) correct even if rational parties prefer to mislead others i.e. $U^{NF} \geq U^{TT}$ and 2) U^{NF}–independent. We also suggest its extension to the (t, n) setting. However, like the protocol of [2], our protocol is dependent on other utility values such as U^{TN}, U^{TT} and U^{NN}. In many scenarios, the act of misleading can be potentially more harmful than the act of selfishness. If a protocol designer wrongly estimates the U^{NF} utility values, the execution of a correct and fair RSS protocol may still result in some of the parties being misled due to the wrongly estimated parameter. Moreover, we believe that estimation of U^{NF} is more difficult than that of U^{TN}, U^{TT} or U^{NN}. The impact of knowing the correct value of a secret is more well–understood than that of believing in a wrong value as the correct one. The existence of a U^{NF}–independent correct and fair rational secret reconstruction protocol is therefore advantageous even if it is dependent on other utility values.

Until now, a general pattern for a rational secret sharing scheme has been the following. Each party gets from the dealer a list of shares, one of which is that of the actual secret and the remaining of fake secrets. The position of this actual share is not known to the players beforehand. This position is chosen according to a geometric distribution $\mathcal{G}(\beta)$, where the parameter β in turn depends on the utility values. In each round of communication, players (either simultaneously or non-simultaneously) broadcast or send individually to each of the other players (in absence of broadcast channel) the current share in its list. The shares are signed by the honest dealer, so no player can give out false shares undetected and the only possible actions in a round are to 1) send the message or 2) remain silent. The round in which the shares of the actual secret are revealed and hence the secret is reconstructed is called the revelation/definitive round. The players are made aware that they have crossed the revelation round by the reconstruction/exchange of an indicator (a bit in [9], a signal in [15]) in the subsequent round. In case of non-simultaneous channels, the indicator cannot be reconstructed/interpreted, as the player who is to communicate last in this round already knows that the round before was the revelation round (because he has the indicator) and quits the protocol immediately without sending messages (shares/signals as the case may be) further. When the deviating player quits, other players also conclude that the secret has been reconstructed in the last round.

Basically, when a party quits in any round, there can be two scenarios: 1) the party quits because it has already obtained the secret and 2) the party quits because it wants others to believe that the secret has been obtained when in reality it is not so. In secret reconstruction protocols for non-simultaneous channels, we see that, whenever a party aborts, the other party assumes that this abortion signifies that the former has obtained its output and hence it also outputs the value obtained in the last round[1]. There is no way for the non-deviating party to verify whether this is actually the revelation round i.e. to find out whether scenario (1) holds or scenario (2). This gives rise to the outcome where one party is misled to believe in a false secret as the actual secret whereas the other party gets nothing. Herein arises the question of correctness of protocol output for fair rational secret reconstruction. The means to restore fairness described so far is fine if it is known that parties have the preference $U^{NF} < U^{TT}$. On the other hand if parties have the preference $U^{NF} \geq U^{TT}$, this way of achieving fairness jeopardizes correctness. [2] achieves the solution to this problem by introducing special fake rounds called completely fake rounds (apart from the normal fake rounds that enable fair secret reconstruction) such that the first player to send a share knows which rounds are the special fake rounds and if the second player, who is unaware of this information, halts to pretend that the end of the list has been reached in any of the completely fake rounds then the first player knows that the other party has cheated. However, this protocol is dependent on the value of U^{NF}. Specifically, with probability α a particular round is a completely fake round and with probability $(1 - \alpha)$ it is not. Then for a player to follow the suggested strategy, it can be easily shown that $\alpha < (U^{TT} - U^{NN})/(U^{NF} - U^{NN})$. The dependence of the correctness of their protocol on the value of α introduces utility-dependence. In comparison, we do not use any such parameter. For our protocol to be correct, we take help of auxiliary information introduced by the protocol designer to allow players to check whether the secret reconstructed by them is correct or not. Since the auxiliary information does not depend on any utility values our protocol is U^{NF}–independent [2].

[1] In fact, this seems to be a widely used concept for restoring fairness when another party aborts prematurely. In his work on Oblivious Transfer, one of the most important cryptographic primitives used in secure computation, Rabin [23] had implicitly suggested this general notion of achieving fairness: the design of a protocol to ensure fairness is such that the very act of aborting by one party should reveal crucial information to the other party which helps it to restore fairness. Gordon who observed this in [24] says that this concept turns out to be very similar to the one they use in their work on complete fairness in secure computation with malicious adversary. Specifically, in their protocol for complete fairness in two-party computation of functions over polynomial-sized domains and without an embedded XOR, if the malicious adversary aborts in any round, then the honest party gets information about the adversary's input in the computation and can compute the value of the function itself, restoring fairness.

[2] Our protocol is only U^{NF}–independent because for maintaining fairness we still use a geometric distribution $\mathcal{G}(\beta)$ where β depends on U^{TN}, U^{TT} and U^{NN}.

Table 1. A Comparison of the Characteristics of Rational Secret Reconstruction Mechanisms

RSS Protocols	Special Preferences	Channel/Dealer Charateristics	Properties
Halpern & Teague (2004) [6]		Simultaneous Broadcast; Online Dealer	Valid for $n \geq 3$; Unconditional
Gordon & Katz (2006) [5]		Simultaneous Broadcast; Online Dealer	Valid for $n \geq 2$; Unconditional
Kol & Naor (two protocols) (2008) [9]	$U^{TT} > U^{NF}$	1)Simultaneous broadcast; 2) Non-Simultaneous Broadcast; Offline Dealer	Fair but not correct for $U^{NF} \geq U^{TT}$ in non-simultaneous case; Unconditional; $(2,2)$, t-out-of-n
Ong et al. (2009) [16]	Majority: Rational; Minority: Honest	Non-Simultaneous Broadcast; Offline Dealer	Unconditional; only 2 rounds of communication
Asharov & Lindell (2010) [2]; two protocols	2) $U^{TT} > U^{NF}$ & $U^{NF} \geq U^{TT}$	1) Simultaneous Broadcast; Online Dealer; 2) Non-simultaneous; Offline Dealer	Complete utility independence for $n \geq 3$; Unconditional; First to achieve both correctness and fairness in non-simultaneous channel (with U^{NF} dependence). Also proved impossibility of fair reconstruction protocol in presence of side information. Proved impossibility of U^{NF} independence in non-simultaneous channel for $(2,2)$ case.
Fuchsbauer et al. (2010) [15]; three protocols	$U^{TT} > U^{NF}$	1) Non-simultaneous, 2) point-to-point, Synchronous 3) Asynchronous; Offline Dealer	$(2,2)$; exactly t-out-of-n; Verifiable Random Function (VRF)
Lysyanskaya & Segal (2010) [21]	$U^{TT} > U^{NF}$	Non-simultaneous, point-to-point, synchronous; Offline Dealer	First fair reconstruction protocol in presence of arbitrary side information; (n,n) case; Use of Time Delayed Encryption (TDE) and VRF
Proposed protocol	$U^{TT} > U^{NF}$ & $U^{NF} \geq U^{TT}$	Non-Simultaneous Broadcast; Offline Dealer	U^{NF} independence; $(2,2)$. (t,n) cases; Use of TDE; Uses protocol generated side information.

Our Contributions. Rational parties preferring to mislead others over everybody knowing the correct output may be quite common. When a piece of secret information is to be revealed, then a rational player who believes that others may have the ability to derive a greater benefit from the information than he can, may decide that it is better to mislead others with wrong information even if that means not getting the correct information himself rather than everyone getting the correct information. However, this scenario has received very little

attention from researchers till now. In this work we propose a new $(2, 2)$ correct and fair rational secret sharing protocol for non-simultaneous channels even if rational parties prefer to mislead and it is in computational strict Nash equilibrium in the presence of protocol-induced auxiliary information. The uniqueness of our protocol is that it is independent of a rational party's utility of misleading. The only other protocol suggested in this scenario [2] is dependent on this utility. We also suggest generalization of our protocol to the (t, n) settings. We allow each party to possess protocol-induced auxiliary information in the form of a checking share to be able to check whether the last round was indeed a revelation round. So even after one party aborts, the other party is armed to check whether he has been misled. This in turn causes no party to have any incentive to deviate from the protocol by aborting arbitrarily, before it has obtained the output. The introduction of auxiliary information has its problems which we combat using the time delayed encryption scheme based on cryptographic memory bound functions as proposed in [21].

Organization of the Paper. The paper is organized as follows: in section 2 we formally introduce the nature of parties and the concepts of fairness and correctness and the role of auxiliary information that we use for further discussions; in section 3 we provide an overview of our protocol, discuss about protocol-induced membership-auxiliary information, checking shares, time delayed encryption and the equilibrium concept used in our protocol and then formally present our protocol for rational secret sharing, followed by an analysis of the protocol. In section 4 we suggest extenstion to (t, n) setting and in section 5, we perform complexity analysis. Finally we conclude in section 6.

2 Preliminaries

2.1 Rational Secret Sharing and the Preference of Rational Players

Shamir's (t, n) secret sharing scheme [11] is used to distribute the shares of a secret among n players such that the secret can be reconstructed only when at least t of them cooperate. In the first phase of such a scheme, called the secret sharing phase, a dealer generates n shares $s_1, \ldots, s_n$ of the original secret s and distributes one share to each of the players. In the next phase, called the secret reconstruction phase, the players exchange their shares. If at least t players cooperate in this phase then the secret can be reconstructed. An adversary controlling less than t players cannot reconstruct the secret. In this scenario, the notion of rational players instead of honest players and players controlled by an adversary was introduced in [6]. They pointed out that if players are rational and have specific preferences such as getting the secret itself and allowing as few others possible know the secret, then no player will ever send his share during the reconstruction phase.

A (t, n) rational secret reconstruction protocol $(\Gamma, \overrightarrow{\sigma})_{t,n}$ (where $\overrightarrow{\sigma}=(\sigma_1, \ldots, \sigma_n)$ denotes the strategies followed by the players) may have different outcomes where an outcome is denoted by $\overrightarrow{o((\Gamma, \overrightarrow{\sigma})_{t,n})}=(o_1, \ldots, o_n)$. The utility function

u_i of each party P_i is defined over the set of possible outcomes of the game and are polynomial in the security parameter k. Thus $U_i^{TN} = u_i(1^k, (o_i = s, o_j = \perp))$, $U_i^{TT} = u_i(1^k, (o_i = s, o_j = s))$ (where $i \neq j$) and so on. Different outcomes of the game may result due to the different preferences of each party. Table 2 [3] describes the possible outcomes and corresponding utilities for t=n=2 and any arbitrary alternative strategy σ_i^{dev} and the suggested strategy σ_i corresponding to a party P_i, $(i = 1, 2)$.

Table 2. Outcomes and Utilities for $(2, 2)$ rational secret reconstruction

P_1's outcome (o_1)	P_2's outcome (o_2)	P_1's Utility $U_1(o_1, o_2)$	P_2's Utility $U_2(o_1, o_2)$
o_1=s	o_2=s	$U_1^{TT}(U_1)$	$U_2^{TT}(U_2)$
o_1=$\perp$	o_2=$\perp$	$U_1^{NN}(U_1^-)$	$U_2^{NN}(U_2^-)$
o_1=s	o_2=$\perp$	$U_1^{TN}(U_1^+)$	$U_2^{NT}(U_2^{--})$
o_1=$\perp$	o_2=s	$U_1^{NT}(U_1^{--})$	$U_2^{TN}(U_2^+)$
o_1=$\perp$	$o_2 \notin \{s, \perp\}$	$U_1^{NF}(U_1^f)$	U_2^{FN}
$o_1 \notin \{s, \perp\}$	o_2=$\perp$	U_1^{FN}	$U_2^{NF}(U_2^f)$

There can be other combinations of the outcomes mentioned in the table, other outcomes and corresponding utilities too but we shall consider only the above. Players have their preferences based on the different possible outcomes. We shall refer to the following preference relationships of a party P_i throughout our paper:

1. $\mathcal{R}_1 : U_i^{TN} > U_i^{TT} > U_i^{NN} > U_i^{FN}$ and $U_i^{NF} \geq U_i^{TT}$
2. $\mathcal{R}_2 : U_i^{TN} > U_i^{TT} > U_i^{NN} > U_i^{FN}$ and $U_i^{NF} < U_i^{TT}$

We call $\{U^{TN}, U^{TT}, U^{NN}, U^{NT}, U^{FN}, U^{NF}\}$ the set of utility types. Since both parties in a reconstruction protocol are considered to have the same preference relation, we can represent the above preference relations (by using utility types in place of particular utility values) respectively as follows:

1. $U^{TN} > U^{TT} > U^{NN} > U^{FN}$ and $U^{NF} \geq U^{TT}$
2. $U^{TN} > U^{TT} > U^{NN} > U^{FN}$ and $U^{NF} < U^{TT}$

2.2 Correctness and Fairness

Let $(\Gamma, \overrightarrow{\sigma})_{2,2}$ be a $(2, 2)$ rational secret reconstruction mechanism. Then, we follow the same definitions of complete fairness and correctness in [2] for the two party scenario:

[3] The notations (e.g., U_1 , U_1^- etc.) in brackets for the last two columns represent the corresponding notations used in [2] and [21].

Definition 1. *(Fairness) A rational secret reconstruction mechanism* $(\Gamma, \overrightarrow{\sigma})$ *is said to be completely fair if for every arbitrary alternative strategy* $\sigma_i^{'}$ *followed by party* P_i, $(i \in \{1,2\})$ *there exists a negligible function* μ *in the security parameter* k *such that the following holds:*

$$Pr[o_i(\Gamma, (\sigma_i^{'}, \sigma_{-i})) = s] \leq Pr[o_{-i}(\Gamma, (\sigma_i^{'}, \sigma_{-i})) = s] + \mu(k)$$

Definition 2. *(Correctness) A rational secret reconstruction mechanism* $(\Gamma, \overrightarrow{\sigma})$ *is said to be correct if for every arbitrary alternative strategy* $\sigma_i^{'}$ *followed by party* P_i, $(i \in \{1,2\})$ *there exists a negligible function* μ *in the security parameter* k *such that the following holds:*

$$Pr[o_{-i}(\Gamma, (\sigma_i^{'}, \sigma_{-i})) \notin \{s, \perp\}] \leq \mu(k)$$

2.3 Utility-Independence

A mechanism $(\Gamma, \overrightarrow{\sigma})$ is said to be independent of a given utility type if it achieves its desired set of properties for any value of that utility type [2]. We define utility-independence as in [2]. We have $U = \left\{U^{TN}, U^{TT}, U^{NN}, U^{NT}, U^{FN}, U^{NF}\right\}$.

Definition 3. *(utility independence, adapted from [2]) Let* $\tilde{U} \in U$ *be a particular utility type and* $U^{'} = \left\{U_i^{TN}, U_i^{TT}, U_i^{NN}, U_i^{FN}, U_i^{NT}, U_i^{NF}\right\}_{i=1}^{n} \backslash \tilde{U}_{i=1}^{n}$ *be a set of polynomial utility functions excluding all* $\tilde{U}_i$ *values. A mechanism* $(\Gamma, \overrightarrow{\sigma})$ *is said to be* $\tilde{U}$*–utility independent if for all polynomial utility functions* $\tilde{U}_{i=1}^{n}$ *for which the elements in* $U = U^{'} \cup \tilde{U}_{i=1}^{n}$ *satisfies a certain preference relationship* $\mathcal{R}$*, it holds that* $(\Gamma, \overrightarrow{\sigma})$ *is a fair reconstruction mechanism for that preference relationship* $\mathcal{R}$ *among the elements of* U*.*

2.4 The Role of Auxiliary Information

[2] discusses the effect of side information possessed by a rational party in a secret reconstruction mechanism. Referring to the secret reconstruction mechanism of [9] they argued that given any auxiliary information about the secret or the access to some membership oracle O that can be queried on whether the current secret $s^{'}$ in the list is the actual secret s, a party possessing a list of fake secrets and the real secret (the long party in the Kol-Naor mechanism) has no incentive to broadcast the secret during the definitive iteration causing the other party not to learn the secret. Prior protocols for secret reconstruction in the rational setting did not allow side information although possession of side information is natural in most practical scenarios. In [2], it has been shown that this limitation is inherent to the non-simultaneous channel assumption. However, recently, the authors in [21] have developed a time delayed encryption scheme based on cryptographic memory-bound functions and using the same have overcome this impossibility result. In this work, we use protocol-induced auxiliary information to allow parties to check whether the secret they reconstruct is a correct one. By 'protocol–induced'we mean that such auxiliary information is a choice of the

mechanism/protocol designer and participants of the protocol have no freedom to choose it.

We adapt the definition of a membership oracle and a fair reconstruction mechanism with membership-auxiliary information given by [2].

Definition 4. *(membership oracle [2]). Let s be the actual secret and one needs to check whether x is same as the actual secret or not. S is the set of all such x. Then, a membership oracle $O : S \to \{0, 1\}$ is defined as follows:*

$$O_S(x) = \begin{cases} 1 & \text{if } x = s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In previous works with auxiliary information, a general case was considered where a party can possess any membership oracle or any side information that enabled it to recognize the secret once it was reconstructed. Our aim is different. When left to themselves, parties may not possess any side information at all or the nature of side information can vary from party to party (some parties may possess incorrect membership oracles). Therefore, the membership oracle that we use must be correct and provided by the protocol itself to the participants.

Definition 5. *(correct membership oracle) A correct membership oracle $O : S \to 0, 1$ is a membership oracle which has the following properties:*

1. *$Pr[O_S(x) = 1] \leq \mu(k)$ for any $x \neq s$ and*
2. *$Pr[O_S(x) = 0] \leq \mu(k)$ for $x = s$.*

where $\mu(k)$ is a negligible function in the security parameter k.

Definition 6. *(protocol-induced membership oracle) A correct membership oracle $O^{\pi}_{q,i}$ provided by the protocol π to its participant P_i, $(i = 1, 2)$ for the qth execution of π is called a protocol-induced membership oracle.*

3 Correct and Fair Reconstruction Mechanism in Non-simultaneous Channel Model

In this section, we first provide a brief sketch of our $(2, 2)$ rational secret sharing protocol. Next we discuss the role of checking share used in our protocol in more details as well as the time delayed encryption scheme and the equilibrium concepts used before the final formal representation of our protocol.

3.1 Sketch of Our Protocol

The main idea behind our protocol is to release the secret gradually, share by share. Each player is given a list of sub-shares, one for the share to be reconstructed in each round. The secret can be reconstructed after sufficient number of these shares have been reconstructed by each party.

The minimum number of rounds r required to generate enough shares so that the secret can be reconstructed is determined by the dealer randomly from a geometric distribution with parameter β. We want β such that

$$\beta < (U^{TT} - U^{NN})/(U^{TN} - U^{NN}).$$

We call this round the revelation round. The dealer therefore has to generate shares of the secret s according to $(r, r+1)$ Shamir's secret sharing scheme so that $r+1$ shares are obtained. If each party possesses r of these shares of the secret (called the reconstruction shares) then they can reconstruct the secret. None of the parties are aware of the value of r.

The dealer randomly chooses one of the $r+1$ shares as the checking share. For each of the remaining shares, sub-shares are generated for each party so that a list of sub-shares for each party is formed. The dealer also generates shares of d fake secrets where d is also chosen from a geometric distribution with parameter β. Therefore a list distributed to a player contains r sub-shares of the shares of the actual secret followed by shares of d fake secrets such that the total number of rounds is $m{=}r{+}d$, the rth round being the revelation round. The fake secrets are required because each party is given the list of shares beforehand to avoid repeated interaction with the dealer. The checking share is distributed separately. The dealer is assumed to be honest and sends the sub-shares digitally signed (information theoretically secure MACs are used).

In each round, players are required to send the sub-share corresponding to the current round in their lists one by one i.e. non-simultaneously. Players are capable of only two actions in a round: send the correct sub-share (if they send an incorrect sub-share then it can be detected and the protocol can be aborted) or remain silent. If in any round a player does not receive a sub-share from the other party then it aborts. We also require that the first round cannot be chosen to be the revelation round; the dealer may send a special abort message if he gets $r{=}1$ and selects r once again. Players are guaranteed to be able to reconstruct the secret if they cooperate and reconstruct all the shares from all the sub-shares available in their lists.

Given the reconstruction shares and the secret, the extra share called the checking share (which is the protocol induced auxiliary information in our case) can be used to determine correctly whether the secret is the correct one. Also, the checking share itself does not reveal any information about the secret. In addition, the checking share acts as an indicator of the revelation round. So, the purpose of the checking share is to achieve correctness. We provide a detailed discussion on protocol-induced auxiliary information and specifically, the checking share in section 3.2. Introduction of the checking share leads to the problem that the party communicating last in any round can use it to identify the actual secret and quit before the other party obtains the secret (this is discussed further in section 3.2). We solve this problem by encrypting each share with the time–delayed encryption scheme introduced in [21] and then generating sub-shares from the encrypted share. A detailed description of this encryption scheme together with how it solves the problem due to introduction of the checking share is given in section 3.3.

Now, the question is whether a rational player P_j will want to deviate in this situation. We shall show in section 3.6 that a player does not gain anything by deviating.

3.2 Protocol-Induced Membership-Auxiliary Information

As mentioned before, we introduce protocol-induced membership-auxiliary information in the form of an extra share of the secret, called the checking share, to check the correctness of the secret reconstructed. A protocol-induced membership oracle (see definition in section 2.3)should not reveal any information about the secret itself i.e. a party should not be able to conclude anything about the secret by simply observing the auxiliary information or by using arbitrary values as input to the oracle. Moreover, given the secret, the oracle must always (except with negligible probability) give the correct decision on whether this input is the actual secret or not. The approach used by us does not benefit a party considering deviation by giving it any additional power in discovering the secret. This is because the additional information is very specific to the particular execution of the secret sharing and reconstruction mechanism and does not impart any information about the secret when used without participating in the protocol.

Let $(\Gamma_f, \overrightarrow{\sigma_f})$ be a fair secret reconstruction mechanism that assumes only $U^{NF} < U^{TT}$. Suppose that in each round r of the secret reconstruction mechanism, P_1 communicates first and P_2 communicates second. At the end of each such round, a value of the form s_r is reconstructed. If one of the parties quit at any round j then the other party is supposed to output the value reconstructed in the previous round i.e. s_{j-1}. Now, let $(\Gamma_{fc}, \overrightarrow{\sigma_{fc}})$ be a fair secret reconstruction mechanism with a protocol-induced membership oracle O_q^{π}. $\overrightarrow{\sigma_{fc}} = (\sigma_{fc,1}, \sigma_{fc,2})$ is a slight modification of $\overrightarrow{\sigma_f} = (\sigma_{f,1}, \sigma_{f,2})$. $\sigma_{fc,i}$ tells party P_i to follow $\sigma_{f,i}$ till an output as defined by $\sigma_{f,i}$ is obtained and then instructs it to query O_q^{π} with the value received in that step to check whether it is the correct one.

Theorem 1. *Let $(\Gamma_{fc}, \overrightarrow{\sigma_{fc}})$ be a $(2,2)$ fair secret reconstruction mechanism with a protocol-induced membership oracle O_q^{π}.Then $(\Gamma_{fc}, \overrightarrow{\sigma_{fc}})$ is also a correct secret reconstruction mechanism.*

Proof. We delay the proof to Appendix A.

For our protocol, we shall consider that each party P_i is given the protocol-induced auxiliary information $aux_q^{\pi,s}$ and the protocol–induced membership oracle $O_q^{\pi,s}$. We note here that because of the presence of our protocol-induced membership-auxiliary information, our protocol cannot tolerate any other auxiliary information that parties may possess themselves (See Appendix D).

Shamir's (1979) (t, n) threshold secret sharing scheme [11] is inherently linked with the protocol–induced membership oracle we use. Shamir's scheme enables one to generate n shares of a secret s such that any t out of these n shares can be used to reconstruct the secret. The dealer chooses a random $t-1$ degree polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}$ where a_0 is set to be equal to

the secret s and the remaining coefficients $a_1, \ldots, a_{t-1}$ are randomly chosen from a uniform distribution over the integers in $[0, p)$ where p is a prime greater than both s and n. The shares are computed as $s_i = f(y_i) mod p$, where $0 < y_i < p$ and $i = 1 \ldots n$.

Now let us consider that the value of t is unknown to Bob who wants to reconstruct a secret from r shares ($r < n$) he has gathered. Therefore, it is completely unknown to him whether he has sufficient shares (i.e. if $r > t$) to reconstruct the secret. Even if he is told that he has sufficient shares, then also he does not know exactly how many of these shares should be used to reconstruct the correct secret. We use this fact to our benefit.

Bob can hold in reserve one of the shares he has and try to reconstruct the secret using different numbers of shares from the remaining shares. After each reconstruction he can use the reserved share to check whether the reconstructed value is the correct secret or not. Specifically on reconstructing a secret $s_r^{'}$ from $r^{'} < r$ shares, he can write the following:

$$f_{r'}(x) = s_{r'} + a_1^{'} x + a_2^{'} x^2 + \ldots + a'_{r'-1} x^{r'-1}$$

Now let us assume that the reserved share s_q is represented as $(y_q, f(y_q) mod p)$.

Claim 1. *If $f_{r'}(y_q) = f(y_q)$, then a player can definitely conclude that $s_{r'} = s$; otherwise it concludes that $s_{r'} \neq s$.*

Proof. For the claim to be true the following two conditions should be fulfilled [by definition of correct membership oracle]:

1. $Pr[f_{r'}(y_q) = f(y_q)] \leq \mu(p)$ for any $s_{r'} \neq s$.
2. $Pr[f_{r'}(y_q) \neq f(y_q)] \leq \mu(p)$ for $s_{r'} = s$.

The second condition always holds by the property of polynomial interpolation. Now, is it possible that even if $s_{r'} \neq s$, $f_{r'}(y_q) = f(y_q)$ holds true? Since $f(x)$ is a randomly chosen polynomial in $[0, p)$, the probability of the point represented by the reserved share lying on both $f(x)$ and $f_{r'}(x)$ where $r^{'} \neq t$ is negligible. So the first condition also holds. Therefore we can conclude that the reserved share (which we call checking share throughout the paper) can serve as a protocol-induced auxiliary information.

Checking Shares. Let us suppose that the player P_1 communicates first in each round whereas the player P_2 communicates last. When P_2 quits in any round then it can have two meanings for P_1: 1) P_2 has already obtained the secret in the last round (i.e. P_2 has the preference $U_2^{NF} < U_2^{TT}$) or 2) P_2 has not obtained the secret but is trying to mislead P_1 in believing that the secret has been obtained in the last round (i.e. P_2 has the preference $U_2^{NF} \geq U_2^{TT}$). By giving a checking share (a share of the actual secret) we enable P_1 to distinguish between scenarios (1) and (2). However, if the checking share is available only to P_1, then P_2 is dependent on the P_1 to know when the revelation round takes place and is thus vulnerable to deviations by P_1. If the checking share is available only to P_2 , then at the end of each round P_2 can check whether it has enough shares to be able

to reconstruct the secret and hence comes to know the revelation round before both parties obtain the secret thereby resulting in an unfair outcome. Therefore the checking share needs to be given to both the parties in such a way that it cannot be used to check whether the current round is the revelation round but can be used to detect if the last round was a revelation round.

The advantage of such a checking share over indicators is that the checking share does not require reconstruction and is readily available to both players whereas indicators are only available to each player if both players send their message in that round. So even when one party aborts prematurely, the other party can check whether the secret reconstructed with the available shares is the correct one by using the checking share. This is not possible with an indicator bit which will not even be reconstructed in the event of one party deviating prematurely. However, the disadvantage is that the checking share acts as auxiliary information that enables to identify the correct secret whereas an indicator bit is in no way related to the correct secret itself. Before sending its share in each round, with the help of this checking share P_2 can check whether it has obtained the actual secret. If it has, then P_2 will quit before sending its message for that round to P_1. Therefore, P_1 will be unable to get the secret leading to an unfair outcome. Therefore it is important to use the checking share in such a manner that it cannot provide any undue advantage to any party in identifying the secret (for example P_2 cannot take the help of the checking share to identify the revelation round before P_1). In [21], the authors have proposed a secret reconstruction mechanism in the standard point-to-point network where parties have auxiliary information. Their protocol develops and uses the concept of Cryptographic Memory-bound Functions which is used in a time delayed encryption scheme to prevent a party from identifying the correct secret before others with the help of the auxiliary information it has. We use the same concept to prevent misuse of the checking share by any party.

We can show that the introduction of the checking share is done without relying on the actual value of U^{NF}.

3.3 Time Delayed Encryption

When players have auxiliary information, then in each round, a deviating player tries to decide whether the current round is the revelation round by checking the reconstructed secret with the auxiliary information. Once the auxiliary information tells this player that the secret has been reconstructed, the player immediately quits without sending its own share. This results in unfairness as the other player cannot reconstruct the secret. A time delayed encryption scheme becomes handy in this situation. A message that has been encrypted by this scheme can only be decrypted after a moderate amount of time has elapsed. Although there has been much work on this type of schemes in the field of time release cryptography, the construction of a time delayed encryption scheme where the time delay is introduced with the help of cryptographic memory bound functions (instead of Time Lock Puzzles [25] that require a huge computational overhead and hence is dependent on CPU speed) was proposed in [21].

A time delayed encryption scheme $(Gen, Enc_K, Dec_K, Unseal_F)$ consists of 1) the algorithm Gen that on input the security parameter 1^k and the hardness parameter h (such that 2^h is a large polynomial in k) outputs a key K, a sealed key K' and some additional information F used to find the key; 2) the encryption and decryption algorithms Enc_K and Dec_K respectively that use the key K and 3) the algorithm $Unseal_F$ such that $Unseal_F(K')$=K. The time delay is introduced by $Unseal_F$ because its running time is lower bounded by $\Omega(2^h)$ i.e. if the reconstructed message is each round is encrypted with this scheme then none of the parties can recover the message in less than $\Omega(2^h)$ steps. Because Cryptographic Memory-Bound Function is used for the construction, these steps are in fact memory accesses i.e. the evaluation of $Unseal_F(\tilde{K})$ requires at least $\Omega(2^h)$ memory accesses.

We use this time-delayed encryption scheme to encrypt the r shares of the secret generated by the dealer (i.e. all shares except the checking share) and then generate sub-shares from the encrypted shares for distribution to the players. This allows the players to reconstruct the encrypted share in each round but does not allow any of them to decrypt the share obtained in the current round till a certain time has elapsed. Each round has to be completed within a certain time limit. If a party does not receive any message for a particular round from the other party within this deadline then it assumes that the other party has quit. If a player wants to decrypt the encrypted share then it has to make a minimum number of memory accesses. The time delay in decryption is such that it causes the party to miss the deadline for sending the message in this round. Therefore a party cannot decide whether the actual secret has been obtained in the current round without missing the deadline which in turn informs the other party of the misbehavior.

We discuss the timing model necessary for fruitfully utilizing the time-delayed encryption scheme in Appendix B.

3.4 Equilibrium Concept

Due to lack of space we defer a discussion on the equilibrium concepts used in the literature of rational secret reconstruction mechanisms to Appendix C. For our protocol we use computational strict Nash Equilibrium in the presence of protocol-induced auxiliary information. We must note that in our case all the players have the same side information denoted by $(aux_q^{\pi,s}, O_q^{\pi,s})$ when induced by the suggested strategy i.e. protocol π.

Definition 7. *(Computational Nash Equilibrium with protocol-induced side information [21]) The suggested strategy σ in the mechanism (Γ, σ) is a computational Nash Equilibrium in the presence of protocol–induced auxiliary information $(aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s})$ if for every P_i any probabilistic polynomial time strategy $\sigma_i^{'}$, $u_i((\sigma_i^{'}, \sigma_{-i}), aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s}) \leq u_i(\sigma, aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s}) + \mu^{'}(k)$ for some negligible $\mu^{'}$.*

Let $\sigma_i^{'} \not\approx (aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s})\sigma$ denote equivalent play (originally defined in [15] and modified in [21] for the case of side information) in the presence of protocol-induced side information. We refer the reader to [21] for detailed discussion.

Definition 8. *(Computational strict Nash Equilibrium with protocol-induced side information [21]) The suggested strategy σ in the mechanism (Γ, σ) is a computational strict Nash Equilibrium in the presence of protocol-induced auxiliary information if it is a Nash Equilibrium with protocol-induced auxiliary information and for every $P_i$for any probabilistic polynomial time strategy $\sigma_i^{'} \not\approx (aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s})\sigma$, $u_i((\sigma_i^{'}, \sigma_{-i}), aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s}) < u_i(\sigma, aux_q^{\sigma,\pi,s}, O_q^{\sigma,\pi,s}) + \mu^{''}(k)$ for some negligible $\mu^{''}$.*

3.5 Our Protocol

In this section, we give the formal description of our protocol. Note that in the description below $(Gen, Enc_K, Dec_K, Unseal_F)$ is the time delayed encryption scheme described in section 4.3.

Protocol ShareGen : The Dealer's Protocol
Inputs. The secret s possessed by the dealer; β, the parameter for the geometric distribution $\mathcal{G}(\beta)$
Computation. The dealer does the following:

1. Generate $r \sim \mathcal{G}(\beta)$.
2. $K_i, K_i^{'}, F_i \leftarrow Gen(1^k), i = 1, \ldots, r$.
3. Use $(r, r+1)$ Shamir's Secret Sharing Scheme to generate r shares of s. Suppose the polynomial used is $f(x)$ where $f(0)$ is set to be equal to the secret s and the remaining coefficients $a_1, \ldots, a_{r-1}$ are randomly chosen from a uniform distribution over the integers in $[0, p)$ where p is a prime number greater than both s and $r+1$. Each share s_i can be represented as $(y_i, f(y_i))$[4] where $0 < y_i < p$ for each $i = 0, \ldots, r$, y_i is chosen randomly.
4. Choose s_{check} to be the 0th share among these $(r+1)$ shares such. Then, s_{check} is of the form $(y_0, f(y_0))$.
5. For each share s_i, $i = 1, \ldots, r$, compute $c_i \leftarrow Enc_{K_i}(s_i)$ and set $c_i^{'} \leftarrow (c_i, K_i^{'})$.
6. For each encrypted share $c_i^{'}$, $i = 1, \ldots, r$, generate sub-shares $c_{i,j}^{'}$ $(j = 1, 2)$ such that $c_i^{'} = c_{i,1}^{'} \oplus c_{i,2}^{'}$.
7. Generate random values $c_{i,j}^{'}$ (for $i = r+1, \ldots, r+d$ and $j = 1, 2$), d is chosen according to the geometric distribution $\mathcal{G}(\beta)$.
8. Construct list $list_j$, $(j = 1, 2)$ to contain $c_{1,j}^{'}, \ldots, c_{r+d,j}^{'}$ for player P_j $(j = 1, 2)$.

Output. Distribute to each player P_j a list $list_j$, $j = 1, 2$. Also distribute the checking share s_{check} to each player.

[4] The accurate way to write is $(y_i, f(y_i) mod p)$. We drop $mod p$ for simplicity of representation.

Protocol Reconstruct: The Players' Protocol
This protocol consists of two phases, the Communication Phase and the Processing Phase. In the Communication Phase players communicate to gather sub-shares, whereas in the Processing Phase players process these sub-shares obtained in the Communication Phase to get the shares of the secret. Thus the Processing Phase for one share works in parallel with the Communication Phase for a subsequent share. An 'abort' in any round of the Communication Phase implies quitting further communication with the other party; however, the aborting party still continues with the processing phase to see whether the secret can be reconstructed from the shares obtained till that round. A 'quit' in the Processing Phase means either the secret has been obtained and hence the next round in the Communication Phase is no longer required or the Communication Phase has been aborted and the shares obtained till the round of abort are not sufficient to reconstruct the secret.

Inputs. List of sub-shares $list_j$ received by each player P_j , $j = 1, 2$ from the dealer.
Communication Phase
In each round, P_1 communicates first.
P_1 communicates first as follows:

1. If in the last round (except if the current round is the first one) P_1 has not received a share within the specified deadline from P_2 or if the share received is not signed properly then abort; else continue till the Processing Phase outputs the secret.
2. Send the current share from $list_1$.
3. Check for shares sent by P_2 till the specified deadline.

P_2 communicates next as follows:

1. If in the current round P_2 has not received a share from P_1 within the specified deadline or if the share received is not signed properly then abort; else continue till the Processing Phase outputs the secret.
2. Send the current share in the list $list_2$.
3. Check for shares sent by P_1 till the specified deadline.

Processing Phase
This phase is carried out by each party on its own in parallel to the communication phase. It can start at least after one round of communication i.e. after the sub-shares of at least one encrypted share of the secret has been gathered by each party.

Until the sub-shares obtained from the Communication Phase is exhausted or until the secret is obtained, each P_j $(j = 1, 2)$ does the following in the ith round of the Processing Phase:

1. Reconstruct $c_i^{'}$ from $c_{i,1}^{'}$ and $c_{i,2}^{'}$.
2. Interpret $c_i^{'}$ as $(c_i, K_i^{'})$.

3. Compute $K_i \leftarrow Unseal_{F_i}(K_i^{'})$ and find $share_i = Dec_{K_i}(c_i)$.
4. If $i > 1$, reconstruct a polynomial $f_i(x)$ of degree $(i-1)$ corresponding to the shares decrypted till the ith round; else move to the first step.
5. Now, s_{check} is $(y_0, f(y_0))$. If $f_i(y_0) = f(y_0)$ then output the constant term $f_i(0)$ of this polynomial as the desired secret and quit. Otherwise, continue. If all sub-shares obtained from the communication round are exhausted and $f_i(y_0) = f(y_0)$ does not hold then output $\perp$.

Output. Either each party outputs the secret s or each party outputs $\perp$.

3.6 Analysis

Theorem 2. *Let our rational secret reconstruction mechanism be denoted by* $(\Gamma, \overrightarrow{\sigma})$. *Then 1) the prescribed strategy* $\overrightarrow{\sigma}$ *of the game* Γ *is in computational strict Nash Equilibrium in presence of protocol-induced auxiliary information; 2) the output obtained by following* $\overrightarrow{\sigma}$ *is correct and 3)* $(\Gamma, \overrightarrow{\sigma})$ *is* U_{NF} *utility–independent.*

Proof. We consider that each share in the lists that the parties receive is signed by the dealer. Therefore neither party can undetectably send a wrong message to the other (since information theoretic MACs are used). In each round, each party either sends the message or chooses not to send it. The point of contention is that the protocol-induced auxiliary information may incentivize a party to deviate by allowing it to check whether the shares obtained till a certain round gives the correct secret or not thus helping in deciding whether to quit or send its share in that round to the other party. We argue that our protocol is a computational strict Nash equilibrium for a party P_i with $U^{TN} > U^{TT} > U^{NN}$ even in the presence of protocol-induced auxiliary information.

Case I. Suppose, P_1 follows the reconstruction protocol whereas P_2 uses an alternate strategy that instructs it to follow the protocol till the qth round. Now if P_2 decides to quit in round $(q+1)$, P_1 aborts and henceforth no exchange of shares takes place. Since P_2 communicates his share following P_1 in each round, P_2 receives the $(q+1)$th sub–share from P_1. However, by the deadline of round $(q+1)$, P_2 cannot decipher his $(q+1)$th share, by the property of time–delayed encryption. If $(q+1) < r$, then P_2 has not gathered enough shares to be able to reconstruct the secret. If $(q+1) > r$ then both parties obtain the secret. The share obtained in the $(q+1)$th round does not help P_2 in any way. Thus, P_2's expected utility of quitting at any round $(q+1)$ is $\delta U_2^{TN} + (1-\delta)U_2^{NN} \leq \beta U_2^{TN} + (1-\beta)U_2^{NN} < U_2^{TT}$ for $r \geq 1$ (where the probability that the secret is reconstructed with $q+1 = r$ shares is given by $\delta = \beta(1-\beta)^{r-1}$), by our choice of β. Note that if P_2 uses its checking share in place of the $(q+1)$th share for the secret reconstruction, then it loses the capability of making sure whether it has obtained the correct secret and hence loses the capability to decide definitely (instead of guessing) whether to quit or not.

Case II. If we assume that P_2 follows the reconstruction protocol whereas P_1 deviates by using a strategy that instructs it to follow the protocol till the qth round and quit immediately after that, then P_1 does not even get the share in the $(q+1)$th round and the same reasoning as Case I also applies here.

What remains to be shown is that the protocol is correct and U^{NF} independent. We first argue that our protocol is a computational strict Nash equilibrium with protocol-induced side information even when $U^{NF} \geq U^{TT}$. Suppose that P_2 quits in the $(q+1)$th round. Then, by the property of the checking share, P_1 instead of outputting a secret formed from all the shares till the qth round can use the checking share to find whether $q+1 > r$. If not, then P_1 outputs a default value. Therefore P_1 can now distinguish between a silent party P_2 with $U_2^{NF} \geq U_2^{TT}$ and a silent party P_2 with $U_2^{TT} > U_2^{NF}$. So a party with $U^{NF} \geq U^{TT}$ gains only U^{NN} due to its deviation whereas if it follows the protocol it gains U^{TT}. Since $U^{TT} > U^{NN}$, the protocol is a computational strict Nash Equilibrium with protocol–induced side information even for $U^{NF} \geq U^{TT}$. Thus we observe that the equilibrium condition is satisfied for any value of U^{NF} as long as $U^{TT} > U^{NN}$ holds. The value of U^{NF} has not been used to introduce the checking share which plays the crucial role in deciding whether the secret obtained till a particular round is correct or not. Therefore, our protocol is U^{NF} independent. Moreover, by the properties of protocol-induced auxiliary information/ membership oracle, the checking share always succeeds in identifying correctly whether a secret is correct or not. Hence our protocol is correct.

4 Generalization to (t, n) Setting

Assuming the presence of non-simultaneous broadcast channel, our protocol can be extended to the (t, n) setting with some modifications. The dealer would need to generate (t, n) sub-shares from each encrypted share (by using (t, n) Shamir's secret sharing) and distribute these sub-shares to the n players. Players would communicate one-by-one in each round. If within the deadline of any round a player obtains less than t shares, he quits. Obviously, we can consider a rushing adversary i.e. the deviating party is the last (i.e. the tth person) to communicate in any round. In that case, this party has to decide whether or not to quit in any round before he is able to decrypt the share he reconstructs from the sub-shares obtained in that round. If he tries to decrypt before taking the decision then, by the property of time-delayed encryption, the deadline for that round is over and all other players quit. So, the same logic as presented in section 3.6 applies here also.

However, if point-to-point network is considered for the (t, n) setting, then the generalization is not easy. In that case, instead of the most general (t, n) setting, we can first look at the (n, n) setting as in[21, 15] or exactly t-out-of-n setting as in [15].

5 Complexity Analysis

In our protocol, the communication phase and the processing phase run parallely. For each player and for each share i to be reconstructed, we need a round of communication phase i.e. CP_i and a round of processing phase PP_i. The processing phase PP_i will coincide in time with CP_{i+1} (since the processing phase must start after one round of communication phase) and overlap partially with CP_{i+2} in time. So the time required for one round of Processing Phase is $(1+\theta)$ times the time required for one round of Communication Phase where θ is chosen by protocol designer and the time delay for the time-delayed encryption scheme should be designed to accomodate θ. The number of rounds for both Communication Phase and Processing Phase is r. So the total elapsed time for Protocol Reconstruct is $(1 + r + \theta)T_{cp}$ where T_{cp} is the time required for one round of Communication Phase. Therefore we are interested on the upper-bound of r.

The size of each list of sub-shares distributed to each player will depend on $r + d$. So we also calculate the upper-bound on $r + d$.

Upper-Bound on r. We have assumed that r is chosen according to a geometric distribution $\mathcal{G}(\beta)$. Also, for a fair rational secret reconstruction protocol, the choice of β is such that

$$0 < \beta < \beta_0 = (U^{TT} - U^{NN})/(U^{TT} - U^{NN}) < 1.$$

Now, given any $\epsilon > 0$ error, we wish to have

$$Pr[r > R] < \epsilon$$
$$i.e. Pr[r > R] = (1-\beta)^R < \epsilon$$
$$i.e. R > \ln \epsilon / \ln(1-\beta)$$

Therefore, we have $Pr[r \leq \lceil \ln \epsilon / \ln(1-\beta) \rceil] > 1 - \epsilon$, where $0 < \epsilon < 1$.

Upper-Bound on $r + d$. We have $r, d \sim \mathcal{G}(\beta)$, r and d are i.i.d random variables, where $0 < \beta < \beta_0 = (U^{TT} - U^{NN})/(U^{TT} - U^{NN}) < 1$.

Given any error $\epsilon > 0$, to have $Pr[r > T/2] < \epsilon/2$, we need

$$T/2 > \ln(\epsilon/2)/\ln(1-\beta)$$

where T is a constant.

This also holds for d.

Now,

$$Pr[r + d > T]$$
$$\leq Pr[r > T/2 or d > T/2]$$
$$\leq Pr[r > T/2] + Pr[d > T/2]$$
$$= 2Pr[r > T/2] < \epsilon,$$

if $T/2 > \ln(\epsilon/2)/\ln(1-\beta)$ or $T > 2\ln(\epsilon/2)/\ln(1-\beta)$.
Therefore,

$$Pr[r + d \leq \lceil 2\ln(\epsilon/2)/\ln(1-\beta)\rceil] > 1 - \epsilon$$

for $0 < \epsilon < 1$.

6 Conclusion

This paper deals with a problem in rational secret sharing that has received very little attention till now. We have proposed a $(2, 2)$ rational secret sharing protocol that is fair and correct as well as independent of the U^{NF}–utility of a rational participant even when $U^{NF} \geq U^{TT}$ in the non-simultaneous channel model and show that it is in computational strict Nash equilibrium in the presence of protocol-induced auxiliary information. We have also given a generalization the protocol to the (t, n) settings.

Acknowledgement. We are indebted to the anonymous reviewers for their numerous useful comments and suggestions. We would like to thank them for their kind efforts to help us improve our work.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: PODC 2006 Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing, pp. 53–62. ACM, New York (2006)
2. Asharov, G., Lindell, Y.: Utility Dependence in Correct and Fair Rational Secret Sharing. Journal of Cryptology 24(1), 157–202 (2010)
3. Dodis, Y., Rabin, T.: Cryptography and Game Theory. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (eds.) Algorithmic Game Theory, pp. 181–205. Cambridge University Press, New York (2007)
4. Goldreich, O.: Foundations of Cryptography Basic Applications, vol. II. Cambridge University Press, Cambridge (2004)
5. Gordon, S.D., Katz, J.: Rational Secret Sharing, Revisited. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 229–241. Springer, Heidelberg (2006)
6. Halpern, J., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: STOC 2004 Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing, pp. 623–632. ACM, New York (2004)
7. Izmalkov, S., Micali, S., Lepinski, M.: Rational secure computation and ideal mechanism design. In: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, pp. 585–594 (2005)
8. Katz, J.: Bridging game theory and cryptography: Recent results and future directions. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 251–272. Springer, Heidelberg (2008)
9. Kol, G., Naor, M.: Games for exchanging information. In: STOC 2008 Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 423–432. ACM, New York (2008)

10. McGrew, R., Porter, R., Shoham, Y.: Towards a general theory of non-cooperative computation. In: TARK 2003 Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 59–71. ACM, New York (2003)
11. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
12. Shoham, Y., Tenneholtz, M.: Non-cooperative computation: Boolean functions with correctness and exclusivity. Theoretical Computer Science 343(1-2), 97–113 (2005)
13. Lysyanskaya, A., Triandopoulos, N.: Rationality and Adversarial Behavior in Multi-party Computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
14. Mas-Collel, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, New York (1995)
15. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient Rational Secret Sharing in Standard Communication Networks. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 419–436. Springer, Heidelberg (2010)
16. Ong, S.J., Parkes, D.C., Rosen, A., Vadhan, S.: Fairness with an Honest Minority and a Rational Majority. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 36–53. Springer, Heidelberg (2009)
17. Kol, G., Naor, M.: Cryptography and Game Theory: Designing protocols for exchanging information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
18. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (2004)
19. Groce, A., Katz, J.: Fair Computation with Rational Players. Cryptology ePrint Archive: Report 2011/396 (2011)
20. Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete Fairness in Secure Two-Party Computation. Journal of the ACM 58(6), Article No. 24 (2011)
21. Lysyanskaya, A., Segal, A.: Rational Secret Sharing with Side Information in Point-to-Point Networks via Time Delayed Encryption. IACR Cryptology ePrint Archive: Report 2010/540. IACR (2010)
22. Dwork, C., Goldberg, A.V., Naor, M.: On Memory-bound Functions for Fighting Spam. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 426–444. Springer, Heidelberg (2003)
23. Rabin, M.O.: How to Exchange Secrets with Oblivious Transfer. Technical Report TR-8, Aiken Computation Lab, Harvard University (1981), `http://eprint.iacr.org/2005/187`
24. Gordon, S.D.: On Fairness in Secure Computation. Ph. D. Thesis. University of Maryland, College Park, USA (2010)
25. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Technical report. Cambridge, MA, USA (1996)

A Correctness in Presence of Protocol-Induced Auxiliary Information

Theorem. *Let* $(\Gamma_{fc}, \overrightarrow{\sigma_{fc}})$ *be a* $(2,2)$ *fair secret reconstruction mechanism with a protocol-induced membership oracle* O_q^{π}. *Then* $(\Gamma_{fc}, \overrightarrow{\sigma_{fc}})$ *is also a correct secret reconstruction mechanism.*

Proof. By assumption, in spite of the presence of a membership oracle with each party, the reconstruction mechanism is fair i.e. none of the parties can identify the revelation round before the other. Now suppose party P_2 has a deviation strategy σ_{dev} that tells it to play according to σ_2 for the first $r^{'}$ rounds and then quit (i.e. remain silent in all rounds henceforth). By assumption, P_1 possesses $O^{\pi}_{q,1}$ and P_2 possesses $O^{\pi}_{q,2}$. Now, if P_2 quits in round $(r^{'}+1)$, then by the suggested strategy $_{fc,1}$, P_1 outputs $s^{'}_r$ if $O^{\pi}_{q,1}(s^{'}_r) = 1$ else it outputs $\perp$. The same argument also holds if P_1 is the deviating party and P_2 the non-deviating party. We have already seen that $u_2(\sigma_1, \sigma_{dev}) = U_2^{NF}$ whereas $u_2(\sigma_1, \sigma_2) = U_2^{TT}$. Since $U_2^{NF} \geq U_2^{TT}$, P_2's best strategy is to follow σ_{dev} rather than the suggested strategy σ_2. So (σ_1, σ_2) is no more an equilibrium strategy. On the other hand, $u_2(\sigma_{fc,1}, \sigma_{dev}) = U_2^{NN}$ while $u_2(\sigma_{fc,1}, \sigma_{fc,2}) = U_2^{TT}$. So $(\sigma_{fc,1}, \sigma_{fc,2})$ is a strictly better strategy profile than $(\sigma_{fc,1}, \sigma_{dev})$ and is an equilibrium strategy whenever there is a party with $U^{NF} \geq U^{TT}$.

B Timing Model

If the time delayed encryption scheme is to be used fruitfully to prevent the misuse of auxiliary information, then it is necessary for each party to know how to find out whether a certain message from another party was received within a given deadline. The timing model for this purpose is discussed in details in [21]. We describe it here very briefly. Both parties must agree on the maximum values for clock drift (τ), network latency (Δ) and speed ($speed_{max}$) of each party. If P_2 is supposed to send a message to P_1 at time t then P_1 must know that if P_2 is following the protocol, then his message must reach P_1 by the time his local clock shows $t + \Delta + \tau$. If any round requires l computation steps scheduled to begin at time t then both parties must have completed the computation by time $t + l/speed_{min}$ where $speed_{min}$ is the minimum of the speeds of both the parties. For our protocol we assume (as in [21]) that the first round of the protocol begins at the pre–decided time t_1. Henceforth, round $q > 1$ starts at $t_q = t_{q-1} + \Delta + \tau + m/speed_{min}$ where m is the maximum number of steps required for computations in each round. So at local time t_q, each party checks whether it can compute s or some party deviated in the last round (i.e. the message from that party for round $q-1$ did not reach till t_q). If the later is true then it quits and moves to the post-processing steps. Each party computes its own message and sends it to the other by time $t_q + m/speed_{min}$.

C Equilibrium Concepts Used in Rational Secret Reconstruction Mechanisms

A rational secret reconstruction protocol should be such that no player has any incentive to deviate from this protocol. Consequently, Nash equilibrium and its several variants have been used as equilibrium concept in the literature of rational secret sharing. A suggested strategy is in Nash equilibrium when given

that everyone else is following the suggested strategy, there is no incentive for a player to deviate from this strategy. However it can be easily shown that even though Shamir's (1979) secret sharing protocol is a Nash equilibrium for $t < n$, there are still strategies that are weakly better than it. This suggests the need for stronger versions of Nash equilibrium to remove such unstable solutions. Again, in the setting of rational secret sharing, in most cases, players are assumed to be polynomial time which calls for a suitable modification in the notion of Nash equilibrium used. Taking such facts into consideration following variants of Nash equilibrium have been used: 1) Nash equilibrium that survives iterated deletion of weakly dominated strategies [6]; 2) strict Nash equilibrium which becomes useful when the payoffs from playing a good strategy and a bad strategy are so close that any minor changes in the beliefs of players about the strategy others are going to adopt may lead each of them to play the bad strategy [9]; 3) computational strict Nash equilibrium [15] where except for non-negligible probability a polynomial time player has a non-negligible loss from deviating; 4) computational Nash equilibrium that is stable with respect to trembles [15] where every other player follows the suggested strategy with high probability; 5) computational strict Nash equilibrium with side information and computational Nash equilibrium with respect to trembles [21] which take into account the fact that each player has access to auxiliary information and a side information oracle.

D Fairness in Presence of Auxiliary Information

The protocol of [21] is fair in spite of the presence of arbitrary auxiliary information. In contrast, our protocol cannot tolerate arbitrary auxiliary information that parties may possess themselves, other than the protocol-induced one. When a party possesses auxiliary information that enables it to identify the actual secret, then it will use the checking share to reconstruct the secret instead of using it for checking purpose. It will then use the auxiliary information it possesses to verify whether the actual secret is obtained. Once it knows this, it will abort early causing other parties to have one share less than required to reconstruct the secret. Thus the protocol will then become unfair.